КОМП'ЮТЕРНІ НАУКИ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

УДК 629.33:681.51 **DOI: 10.30977/BUL.2219-5548.2021.94.0.142**

СИСТЕМА ДЛЯ АНАЛІЗУ ВЕЛИКИХ МАСИВІВ ДАНИХ ЗА ДОПОМОГОЮ АЛГОРИТМІВ МАШИННОГО НАВЧАННЯ

Пронін С. В., Мірошниченко М. О.
Харківський національний автомобільно-дорожній університет

Анотація. *У статті розглядаються інструменти для створення систем машинного навчання й аналізу даних. Розглянуто основні методи машинного навчання, проаналізовано інструментарій для побудови систем аналізу даних.*
Ключові слова: *машинне навчання, аналіз даних, нейронна мережа, дерево рішень, scikit-learn.*

Вступ

Світовий обсяг цифрованої інформації зростає по експоненті. За даними компанії IBS, до 2003 р. світ накопичив 5 ексабайт. До 2008 р. цей обсяг зріс до 0,18 зетабайт до 2011 р. – до 1,76 зетабайт, до 2013 р. – до 4,4 зетабайт. У травні 2015 р. глобальна кількість даних перевищила 6,5 зетабайт. На 2020 р., за прогнозами, людство сформує 40–44 зетабайт інформації.

За розрахунками IBS, 2013 р. тільки 1,5 % накопичених масивів даних мало інформаційну цінність. У зв'язку з цим активно розвиваються технології обробки великих даних. Вони дозволять структурувати інформацію та отримати з цього користь. Аналіз великих даних дозволить побачити приховані закономірності, непомітні обмеженому людському сприйняттю. Це дає безпрецедентні можливості оптимізації всіх сфер нашого життя: державного управління, медицини, телекомунікації, фінансів, транспорту, виробництва тощо.

Методи машинного навчання застосовуються в різноманітних галузях і допомагають вирішувати безліч завдань: від виявлення спаму і актів шахрайства до розпізнавання й генерації зображень і музичних композицій.

Ця стаття розглядає одну з імовірних ідей застосування парадигми великих даних – можливість на основі загальнодоступних даних створювати програми для оцінки переваг окремих користувачів. Успішне вирішення такого завдання може бути використано, зокрема, у сфері інтернет-комерції – знання про те, який товар краще для покупця, допомагають ефективніше організувати контекстні пропозиції товарів, що сприяє збільшенню ефективності бізнесу загалом.

Аналіз публікацій

Технологія машинного навчання на основі аналізу даних бере початок 1950 р., коли почали розробляти перші програми для гри в шашки. За минулі десятиліття загальний принцип не змінився. Зате завдяки бурхливому зростанню обчислювальних потужностей комп'ютерів багаторазово ускладнилися закономірності й прогнози, створювані ними, і розширилося коло проблем і завдань, що вирішуються з використанням машинного навчання [1].

Щоб запустити процес машинного навчання, для початку необхідно зібрати інформацію та сформувати інформаційний масив, придатний для алгоритму, який буде обробляти запити. Процес навчання триває і після виданих прогнозів. Чим більше даних ми проаналізували програмою, тим більш точно вона розпізнає потрібні зображення.

Таким чином основна ідея машинного навчання полягає в тому, щоб навчити комп'ютер «вчитися», тобто виокремлювати з будь-яких даних корисні знання.

У зв'язку з швидким зростанням обсягу інформації виникає гостра необхідність в обробленні та структуруванні цієї інформації для її подальшого використання в навчанні моделі, яка на виході буде давати результат. Першим етапом побудови моделі класифікації є збір і попередня обробка даних – цей процес знаходиться на стику таких розділів як: Великі дані (Big data) [2] та інтелектуальний аналіз даних (Data Mining) [3].

Big data або Великі дані – це загальна назва для великих масивів даних і методів їхньої обробки. Такі дані ефективно обробляються за допомогою масштабованих програмних інструментів, які з'явилися в кінці 2000-х рр. і стали альтернативою традиційних баз даних і рішень Business

Intelligence [4]. Аналіз великих даних проводять для того, щоб отримати нову, раніше невідому інформацію. Подібні відкриття називають інсайтом, що означає осяяння, здогад, раптове розуміння.

Концепція великих даних на сьогодні використовується для таких операцій:

- обробка великих порівняно зі «стандартними» обсягами даних;
- уміння працювати зі швидко динамічними даними в дуже великих обсягах, тобто в ситуаціях постійного зростання обсягів даних, які потрібно структурувати та аналізувати;
- уміння працювати зі структурованими й погано структурованими даними паралельно в різних аспектах.

Великі дані припускають, що на вхід алгоритми отримують потік не завжди структурованої інформації і що з нього можна витягти більше ніж одну ідею.

Для процесу інтелектуального аналізу даних існують цілком певні стандарти процесу, наприклад Cross-Industry Standart Process for Data Mining (CRISP-DM) або SEMMA. Загалом обидва цих стандарти схожі, за винятком, можливо, іменування етапів роботи [5].

Стандарт передбачає шість глобальних фаз процесу аналізу даних:

1. Business Understanding – первісна фаза – спрямована на розуміння поставленого завдання з погляду бізнесу. Також на цій фазі формується проблема аналізу даних, що вирішуватиметься, ставляться завдання, які будуть виконані в процесі досягнення бізнес-мети.

2. Data Understanding – фаза фокусується на первинному аналізі даних щодо проблем їхнього збору, проблем якості даних. Також у цій фазі робляться початкові припущення про способи аналізу, характер прихованих законів тощо.

3. Data Preparation – фаза покриває процеси отримання початкової вихідної інформації, трансформування її в підсумкову вибірку, яка буде подана на вхід моделям аналізу. Проводяться як відбір ознак, так і рішення проблем неякісних даних – відновлення пропусків і позбавлення від викидів у даних.

4. Modeling – фаза описує використання різних методик побудови моделей аналізу, а також процеси налаштування параметрів моделей для досягнення оптимального результату. Процес вибору моделі для розв'язання прикладної задачі аналізу користувачів, а також теоретичні положення побудови моделі та її налаштування будуть описані далі в поточному розділі.

5. Evaluation – ця фаза зосереджується на оцінюванні результатів моделювання з погляду аналізу даних. Перевіряються всі положення й теорій, які використовуються в процесі аналізу, наводяться критерії успішного вирішення бізнес-завдання. Поставлені в фазі Business Understanding цілі мають бути досягнуті.

Огляд методів та алгоритмів класифікації

У штучному інтелекті й машинному навчанні завдання поділу безлічі спостережень (об'єктів) на групи, звані класами, на основі аналізу їхнього формального опису отримало назву – «завдання класифікації». Для класифікації кожна одиниця спостереження належить до певної групи або номінальної категорії на основі деяких властивостей.

Опишемо класичну задачу класифікації. Нехай X – безліч описів об'єктів, Y – кінцева множина номерів (імен, міток) класів. Існує невідома цільова залежність – відображення у $X \to Y$, значення якої відомі тільки на об'єктах кінцевої навчальної вибірки $X_m = (x_1, y_1), ..., (x_m, y_m)$. Потрібно побудувати алгоритм $X \to Y$, здатний класифікувати довільний об'єкт $x \in X$.

У математичній статистиці завдання класифікації називаються також завданнями дискримінантного аналізу.

У машинному навчанні завдання класифікації вирішується з використанням навчання з учителем, оскільки класи визначаються заздалегідь і для прикладів навчальної множини мітки класів задані. Аналітичні моделі, вирішуючи задачу класифікації, називаються класифікаторами.

Завдання класифікації є ще однією з базових задач прикладної статистики та машинного навчання, а також штучного інтелекту загалом. Це пов'язано з тим, що класифікація є одним із найбільш зрозумілих і простих для інтерпретації технологій аналізу даних, а правила класифікації можуть бути сформульовані природною мовою.

До поширених методів вирішення задачі класифікації належать:

- нейронні мережі;
- логістична і пробіт-регресія;
- дерева рішень;
- метод найближчого сусіда;
- метод опорних векторів;
- дискримінантний аналіз;

– баєсівська (наївна) класифікація.

У статті для класифікації ознак були обрані нейронні мережі та дерева прийняття рішень.

Дерево прийняття рішень (також може називатися деревом класифікації або регресійним деревом) – засіб прийняття рішень, що використовується в машинному навчанні, аналізі даних і статистиці. Структурою дерева є «листя» і «гілки» (рис. 1).

На ребрах («гілках») дерева рішення записані атрибути, від яких залежить цільова функція, у «листі» записані значення цільової функції, а в інших вузлах – атрибути, за якими розрізняються випадки. Щоб класифікувати новий випадок, треба спуститися по дереву до листа і видати відповідне значення [3]. Подібні дерева рішень широко використовуються в інтелектуальному аналізі даних. Мета полягає в тому, щоб створити модель, яка передбачає значення цільової змінної на основі декількох змінних на вході.

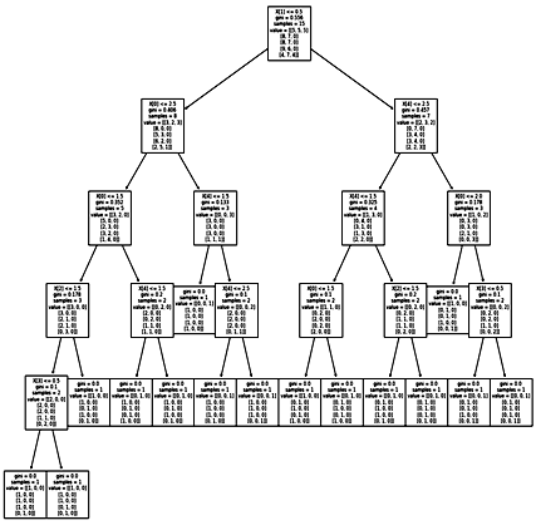

Рис. 1. Загальний вигляд дерева прийняття рішень

Випадковий ліс (англ. Random forest) – алгоритм машинного навчання, що полягає у використанні комітету (ансамблю) розв'язувальних дерев [3]. Алгоритм поєднує в собі дві основні ідеї – метод беґґінґа і метод випадкових підпросторів – і працює таким чином. Нехай дана навчальна вибірка D розміру n. Генерується m нових вибірок Di розміру n ', вибором з D випадково з поверненням. Деякі спостереження можуть потрапити у вибірку кілька разів, деякі можуть взагалі не потрапити. Якщо n ' = n і n велике, то частка різних спостережень у Di буде (1−1/e) ≈ 63,2 %. Далі навчається m класифікаторів на кожній вибірці Di. Для класифікації нової точки ці класифікатори голосують і зараховують точку до класу, за який проголосувала більшість. У методі випадкових підпросторів (random subspace method, RSM) класифікатори навчаються на різних підмножинах ознакового опису, які також виокремлюються випадковим чином. Розглянемо алгоритм побудови випадкового лісу. Нехай навчальна вибірка складається з N прикладів, розмірність простору ознак дорівнює M, і заданий параметр m (у задачах класифікації зазвичай $m \approx \sqrt{M}$). Всі дерева комітету будуються незалежно один від одного за такою процедурою:

1. Згенеруємо випадкову підвибірку з повторенням розміром n з навчальної вибірки. (Отже, деякі приклади потраплять до неї кілька разів, а приблизно N/3 прикладів не увійдуть у неї взагалі.)

2. Побудуємо вирішальне дерево, яке класифікує приклади цієї підвибірки. До того ж під час створення чергового вузла дерева будемо вибирати ознаку, на основі якої проводиться розбиття, не з усіх M ознак, а лише з m випадково вибраних. Вибір найкращої з цих m ознак може здійснюватися різними способами. В оригінальному коді Бреймана використовується критерій Джині. У деяких реалізаціях алгоритму замість нього застосовується критерій приросту інформації.

3. Дерево будується до повного вичерпання підвибірки. Класифікація об'єктів проводиться шляхом голосування: кожне дерево комітету зараховують до одного з класів, і перемагає клас, за який проголосувала найбільша кількість дерев.

Нейронні мережі – це один із напрямів досліджень у сфері штучного інтелекту, оснований на спробах відтворити нервову систему людини, а саме: здатність нервової системи навчатися і виправляти помилки, що має дозволити змоделювати, хоча і досить грубо, роботу людського мозку.

Здатність до моделювання нелінійних процесів, роботи із зашумленими даними і адаптивність дають змогу застосовувати нейронні мережі для вирішення широкого класу завдань. В останні кілька років на основі нейронних мереж було розроблено багато програмних систем, які охоплюють найрізноманітніші галузі: розпізнавання образів, обробка зашумлених даних, доповнення образів, асоціативний пошук, класифікація, оптимізація, прогноз, діагностика, оброблення

сигналів, абстрагування, управління процесами, сегментація даних, стиснення інформації, складні зображення, моделювання складних процесів, машинний зір, розпізнавання мови [3].

Як нейронну мережу обрано багатошаровий перцептрон. Ця нейронна мережа, незалежно від своєї простоти, дозволяє досить ефективно вирішувати більшість завдань, пов'язаних із класифікацією, обробкою, розпізнаванням образів.

Багатошаровий перцептрон (MLP). Нейронна мережа на основі MLP є системою з пов'язаних між собою шарів нейронів. Кожен нейрон характеризується функцією активації, яка перетворює вхідний сигнал нейрона у вихідний. Зв'язки нейронів з іншими нейронами характеризуються коефіцієнтами – так званими вагами зв'язку. Важливим фактором у навчанні нейронної мережі є вид вхідних даних. Для досягнення найкращих результатів необхідно попередньо провести відображення даних на діапазон [-1;1] за допомогою операцій центрування і масштабування (1).

$$x_n = \frac{(x - m_x)}{m_x}. \qquad (1)$$

Процес навчання є ітеративною послідовністю операцій розрахунку вихідного сигналу мережі та подальшого коректування ваг зв'язків. Як алгоритм коригування ваг у мережах на основі MLP зазвичай застосовується алгоритм зворотного поширення помилки.

Алгоритм зворотного поширення помилки (Back propagation algorithm). Алгоритм зворотного поширення помилки – популярний алгоритм навчання плоскошарових нейронних мереж прямого поширення (багатошарових персептронів). Належить до методів навчання з учителем, тому вимагає, щоб у навчальних прикладах були задані цільові значення. Також є одним із найбільш відомих алгоритмів машинного навчання.

Цей алгоритм належить до класу градієнтних алгоритмів, тобто зміни ваг зв'язків виробляються в напрямку мінімізації градієнта помилки. Помилка прогнозу в процесі навчання дорівнює різниці сигналу на виході мережі й еталонного значення виходу, відповідного вхідним даним (2).

$$e_i = (y_i - d_i). \qquad (2)$$

Навчання мережі необхідно виконувати доти, доки середня величина помилки за одну епоху навчання зменшується. Подальше навчання зазвичай приводить до погіршення аналітичних можливостей нейронної мережі. Навчання мережі відбувалося за допомогою алгоритму зворотного поширення помилки та алгоритму градієнтного спуску.

В основі ідеї алгоритму лежить використання вихідної помилки нейронної мережі (3) для обчислення величин корекції ваг нейронів у її прихованих шарах:

$$E = \frac{1}{2}\sum_{i=1}^{k}(y - y')^2, \qquad (3)$$

де k – число вихідних нейронів мережі; y – цільове значення; y' – фактичне вихідне значення.

Алгоритм є ітеративним і використовує принцип навчання «по кроках» (навчання в режимі on-line), коли ваги нейронів мережі коригуються після подачі на її вхід одного навчального прикладу.

На кожній ітерації відбувається два проходи мережі – прямий і зворотний. На прямому вхідний вектор поширюється від входів мережі до її виходів і формує певний вихідний вектор, що відповідає поточному (фактичному) стану ваг. Потім обчислюється помилка нейронної мережі як різниця між фактичним і цільовим значеннями. На зворотному проході ця помилка поширюється від виходу мережі до її входів, і проводиться корекція ваг нейронів за формулою (4):

$$\Delta w_{j,i}(n) = \frac{-\eta\,\partial E_{av}}{\partial w_{ij}}, \qquad (4)$$

де $w_{j,i}$ – вага і-го зв'язку j-го нейрона; η – параметр швидкості навчання, який дозволяє додатково керувати величиною кроку корекції; Δw_{ji} з метою більш точного налаштування на мінімум помилки й підбирається експериментально в процесі навчання (змінюється в інтервалі від 0 до 1).

Стохастичний градієнтний спуск. Цей алгоритм належить до оптимізаційних алгоритмів та використовується для налаштування параметрів моделі машинного навчання. За умови стандартного (або «пакетного», «batch») градієнтного спуску для коригування параметрів моделі використовується градієнт. Градієнт зазвичай вважається як сума градієнтів, викликаних кожним елементом навчання. Вектор параметрів змінюється в напрямку антиградієнта із заданим кроком. Тому стандартному градієнтному спуску потрібно один прохід по навчальних

даних до того, як він зможе змінювати параметри. У разі стохастичного (або «оперативного») градієнтного спуску значення градієнта апроксимується градієнтом функції вартості, обчисленим тільки на одному елементі навчання. Потім параметри змінюються пропорційно наближеному градієнту. Таким чином параметри моделі змінюються після кожного об'єкта навчання.

Для великих масивів даних стохастичний градієнтний спуск може дати значну перевагу у швидкості порівняно зі стандартним градієнтним спуском. Між цими двома видами градієнтного спуску існує компроміс, званий іноді «mini-batch». У цьому випадку градієнт апроксимується сумою для невеликої кількості навчальних зразків.

Як інтерпретацію результату в мережі застосовують функцію softmax та Relu. Softmax – це логістична функція для багатовимірного випадку. Функцію застосовують не до окремого значення, а до вектора. Наприклад, її можна використовувати в тому випадку, коли стоїть завдання багатокласової класифікації.

Для такої класифікації мережу будують таким чином, що на останньому шарі кількість нейронів виявляється рівною кількості шуканих класів. У цьому випадку кожен нейрон має видавати значення ймовірності належності об'єкта до класу (5), тобто значення між нулем і одиницею, а всі нейрони в сумі мають дати одиницю.

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{k=1}^{k} e^{z_k}}. \qquad (5)$$

Relu – Rectified linear unit (ReLU) або «випрямляч» (rectifier, за аналогією з однопівперіодним випрямлячем в електротехніці) є найбільш часто використовуваною функцією активації з 2015 р. Це проста умова і має переваги перед іншими функціями. Функція визначається такою формулою:

$$f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases}. \qquad (6)$$

Застосування методів класифікації, відбору ознак

На сьогодні існує декілька мов програмування, за допомогою яких можна розробляти застосунки для машинного навчання та аналізу даних. Із цих мов можна виокремити мову Python [6].

Одна з основних причин, чому Python використовується для машинного навчання, полягає в тому, що в нього є безліч фреймворків, які спрощують процес написання коду і скорочують час на розробку. Обговоримо, які саме бібліотеки і фреймворки Python використовуються в машинному навчанні. У наукових розрахунках застосовується Numpy, SciPy, у добуванні й аналізі даних – SciKit-Learn. Ці бібліотеки часто використовують разом із TensorFlow, CNTK і Apache Spark, за допомогою яких проектуються нейронні мережі. Крім того, простий синтаксис мови Python допомагає розробнику тестувати складні алгоритми з мінімальною витратою часу на їхню реалізацію.

Для рішення задачі був обраний фреймворк Scikit-learn [7].

Scikit-learn побудована на основі стека SciPy (Scientific Python), який містить:

- NumPy – додає підтримку великих багатовимірних масивів і матриць, а також бібліотеку високорівневих математичних функцій для операцій з ними;
- SciPy – відкрита бібліотека високоякісних наукових інструментів для мови програмування Python;
- Matplotlib – бібліотека для візуалізації двовимірної та тривимірної графіки;
- IPython – інтерактивна оболонка для мови програмування Python, яка надає розширену інтроспекцію, додатковий командний синтаксис, свертку коду й автоматичне доповнення Sympy – бібліотеки для роботи із символьними обчисленнями;
- Pandas реалізує різні структури даних і аналіз.

Бібліотека scikit-learn складається з 35 модулей, які можна поділити на модулі кластеризації, модулі оцінювання моделі й кількісного визначення якості прогнозів, модулі роботи з наборами даних (передобробка, нормалізація), модулі роботи з ознаками (витяг і виявлення найбільш значущих), модулі, що реалізують особисті алгоритми вирішення задач класифікації та регресії.

Кожний модуль складається з класів і функцій та виконує такі завдання:

- кластеризація (Clustering) – угрупування нерозмічених даних;
- перехресна перевірка (Cross Validation) – оцінювання ефективності роботи моделі на незалежних даних;
- набори даних (Datasets) – для зберігання тестових наборів даних і для генерації наборів даних із певними властивостями для дослідження поведінкових властивостей моделі;

- скорочення розмірності (Dimensionality Reduction) – набір алгоритмів для зменшення кількості атрибутів для візуалізації та відбору ознак (Feature Selection), наприклад метод головних компонент (Principal Component Analysis);
- алгоритмічні композиції (Ensemble Methods) – набір методів для комбінування прогнозів декількох моделей;
- витяг ознак (Feature Extraction) – процес визначення атрибутів у даних;
- відбір ознак (Feature Selection) – набір алгоритмів для виявлення значущих атрибутів, на основі яких буде побудована модель;
- оптимізація параметрів алгоритму (Parameter Tuning) – методи для отримання максимально еф-ність віддачі від моделі;
- множинне навчання (Manifold Learning) – підхід нелінійного скорочення розмірності даних.

Окремо потрібно виокремити методи, що реалізують навчання з учителем (Supervised Models). Цей набір методів передбачає:

- узагальнені лінійні моделі (Generalized Linear Models);
- методи дискримінантного аналізу (Discriminate Analysis);
- наївний баєсівський класифікатор (Naive Bayes);
- нейронні мережі (Neural Networks);
- метод опорних векторів (Support Vector Machines):
- дерева прийняття рішень (Decision Trees).

Після аналізу та попередньої обробки даних іде етап застосування методів машинного навчання. У межах цієї статті використовувалися такі методи: випадковий ліс (RF), класифікація з допомогою нейронних мереж (MLP). Як вхідні дані був застосований архів автомобілів Car Evaluation. Car Evaluation – це датасет, дані з якого трансформуються в pandas DataFrame, на його базі тренується модель. Вхідна вибірка є даними з п’яти вхідних параметрів:

– вартість авто;
– вартість обслуговування;
– кількість дверей;
– кількість пасажирів;
– об’єм багажника;
– безпека авто.

На виході нейронної мережі маємо чотири класи, які розраховуються на основі вхідних параметрів:

– актуальне авто;
– неактуальне авто;
– гарне авто;
– дуже гарне авто.

Нижче для кожного класифікатора представлена сітка з множин гіперпараметрів. Вона використовувалася для всіх підходів підвищення якості, і по ній було здійснено пошук оптимального набору гіперпараметрів, що забезпечує максимальне середнє значення збалансованої метрики якості accuracy.

Нижче наведені параметри моделей.

Випадковий ліс:
n_estimators: 10;
criterion: gini;
max_depth: None;
min_samples_split: 2;
min_samples_leaf: 1;

Багатошаровий перцептрон:
hidden_layer_sizes: (64, 64, 4);
activation: relu, relu, softmax;
optimizer: sgd;
loss: categorical_crossentropy;
metrics: accuracy;

У табл. 1 вказано якість методів за п’ятиблоковою перехресною перевіркою після знаходження оптимальних гіперпараметрів. Перехресна перевірка показує адекватність методу загалом щодо даних.

Таблиця 1 – Порівняльна точність алгоритмів

	Accuracy	Loss
RF	0.9602	0.0808
MLP	0.9335	0.0675

Графік залежності точності моделі й кількості втрат від кількості епох навчання зображений нижче (рис. 2).

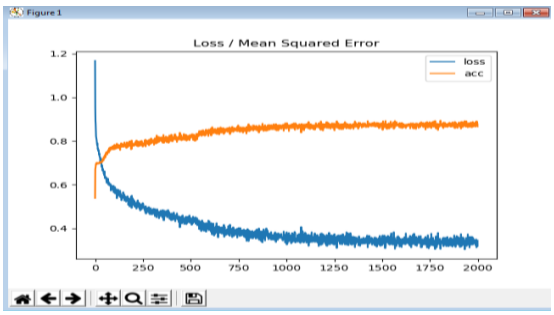

Рис. 2. Графік залежності точності моделі й кількості втрат від кількості епох навчання

Висновок

Розглянуто підхід щодо створення систем машинного навчання та аналізу даних. Для побудови системи були використані

спеціалізовані бібліотеки мови програмування Python – Pandas та Scikit-learn. Серед методів для аналізу були застосовані багатошаровий перцептрон та випадковий ліс, які містяться в складі Scikit-learn. Випадковий ліс порівняно з багатошаровим перцептроном має кращу якість за перехресною перевіркою та більш стійкий щодо гіперпараметрів, та показав менший щодо нейронної мережі час на навчання. Із цього випливає, що для класифікації більш складних даних варто скористатися випадковим лісом.

Література

1. Weka: Data Mining. URL: https://www.cs.waikato.ac.nz/~ml/weka/ (дата звернення: 23.11.2020).
2. Big Data. URL: https://www.it.ua/ru/knowledge-base/technology-innovation/big-data-bolshie-dannye (дата звернення: 23.11.2020).
3. Барсегян А.А., Куприянов М.С. Методы и модели анализа данных. – Санкт-Петербург: БХВ Петербург. – 2004. – 134 с.
4. BI – бизнес-аналитика. URL: https:// www.it.ua/ru/knowledge-base/technology-innovation/business-intelligence-bi (дата звернення: 23.11.2020).
5. Shearer C. The CRISP-DM model: the new blueprint for data mining. – JData Warehousing. – 2000. – С. 22.
6. Documentation Python.org. URL: https://www.python.org/
7. Scikit-learn. Machine Learning in Python. URL: http://scikit-learn.org (дата звернення: 12.11.2020).

References

1. Weka: Data Mining Retrived. URL: https://www.cs.waikato.ac.nz/~ml/weka/ (accessed: 23.11.2020).
2. Big Data Retrived. URL: https://www.it.ua/ru/knowledge-base/technology-innovation/big-data-bolshie-dannye (accessed: 23.11.2020).
3. Barsegyan A.A. Metody` i modeli analiza danny`kh [Data Analysis Methods and Models]. Sankt-Peterburg: BKhV Peterburg. 2004. 134 s.
4. BI – бизнес-аналитика Retrived. URL: https://www.it.ua/ru/knowledge-base/technology-innovation/business-intelligence-bi (accessed: 23.11.2020).
5. Shearer C. The CRISP-DM model: the new blueprint for data mining. JData Warehousing. 2000. C. 22.
6. Documentation Python.org Retrived. URL: https://www.python.org/ (accessed: 23.11.2020).
7. Scikit-learn. Machine Learning in Python. Retrived. URL: http://scikit-learn.org (accessed: 23.11.2020).

Пронін Сергій Вікторович, к.т.н., доцент кафедри комп'ютерних технологій і мехатроніки, psv59777@gmail.com, тел.: 057-707-37-43

Мірошниченко Михайло Олександрович, студент, ХНАДУ,
mikhail.miroshnichenko18@gmail.com,
тел.: 063 047 5641

Харківський національний автомобільно-дорожній університет, 61002, Україна, м. Харків, вул. Ярослава Мудрого, 25.

A system for analyzing large data sets using machine learning algorithms

Abstract. *One of the possible ideas of applying the big data paradigm is considered – the ability to create programs based on publicly available data to assess the preferences of individual users. A successful solution to this problem can be used, in particular, in the field of e-commerce – the knowledge of which product is the best for the buyer creates effective organization of the contextual offerings of products, which in turn leads to increased efficiency of the business as a whole.* ***Problem****. Due to the rapid growth of information, there is an urgent need to process and structure this information for further use in teaching a model that will yield results. The* ***goal*** *is the development of big data analysis systems which is carried out to obtain new information. The* ***methodology*** *of application of machine learning algorithms is used, namely artificial neural network and random forest. As a* ***result****, it was found that the multilayer perceptron gave a better cross-check quality compared to the random forest and was more stable in terms of hyperparameters; and the random forest had less time to learn. It follows that a neural network should be used to classify more complex data. The* ***originality*** *of the work lies in the use of specialized machine learning libraries to create data analysis systems. The* ***practical value*** *of the work is the possibility of creating data analysis systems built using specialized machine learning libraries.*

Key words: *machine learning, data analysis, neural network, decision tree, scikit-learn.*

Pronin Sergey Viktorovich, Ph.D., Associate Professor of Computer Technology and Mechatronics, psv59777@gmail.com,
tel.: 057-707-37-43

Miroshnichenko Mykhailo Oleksandrovych, student of KhNADU
mikhail.miroshnichenko18@gmail.com,
tel.: 063 047 5641

Kharkiv National Automobile and Road University, 61002, Ukraine, Kharkiv, street Yaroslav the Wise, 25.